A Decentralized Digital Voting System Based On Blockchain Architecture

Kaivalya Hemant Mendki
Department of Information Technology Engineering
Walchand College of Engineering, Sangli, India.
kmendki98@gmail.com

Abhishek Rajendra Kotkar
Department of Information Technology Engineering
Walchand College of Engineering, Sangli, India.
kotkarabhishek17@gmail.com

Aditya Pramod Bonde
Department of Information Technology Engineering
Walchand College of Engineering, Sangli, India.
adityabonde19@gmail.com

Under the Guidance of,

Dr. D. B. Kulkarni
Department of Information Technology Engineering
Walchand College of Engineering, Sangli, India.
dinesh.kulkarni@walchandsangli.ac.in

Abstract*— Large section of demographic today do not trust their government.This makes an election a very crucial aspect in a modern democracy. The issue with the current Indian election system is that it can be manipulated and hence is not reliable upto the large extent. Hence cryptographic techniques are employed to ensure the security of voting systems in order to increase its wide adoption. However, in such electronic voting systems, the public bulletin board that is hosted by the third party for publishing and auditing the voting results should be trusted by all participants. Recently, a number of digital solutions have been proposed to address this issue. However, these systems are impractical to use due to the limitations on the voter and candidate numbers supported, and their security techniques which are highly dependent on client-server architecture. To deal with all these issues, we propose practical platform-independent secure and distributed voting system which is developed on top the blockchain architecture. The proposed system hopes to wipe out the part of trust from an election to make it more secure and transparent. Undeniable nature is characteristically given by the fundamental blockchain design, and cryptographic methods like CSPRNG, salt hashing, proof of work are employed to enhance data security measures. There are many substantial social benefits to using the system as well such an easier and quicker voting process which will lead to higher voter turnout. We might definitely see a future where our country has implemented a system similar to ours.

Keywords—evoting, blockchain, decentralized, encryption, consensus, byzantine fault tolerance, P2P.

I. INTRODUCTION

Voting plays a significant role in a democratic society. Almost every local authority allots a significant amount of budget on providing a more robust and trustworthy voting system. For voting systems based on the bulletin, one of the major concerns is whether the voting result that is published on the bulletin can be trusted. Blockchain[1] with the growing popularity and remarkable success in cryptocurrency provides a new paradigm to achieve the public verifiability in such electronic voting systems.

Blockchain was first introduced by Satoshi Nakamoto (a pseudonym)[2], who proposed a peer-to-peer[3] payment system that allows cash transactions through the Internet without relying on trust or the need for a financial institution. Blockchain is secure by design, and an example of a system with a high byzantine failure tolerance.

In a blockchain-based systems there is no trusted centralised coordinator; instead, each node that is involved in the blockchain system holds the data block locally. Based on the assumption that the decentralized consensus protocol is secure and a sufficiently large proportion of blockchain network nodes are honest, the blockchain can be thought of as a conceptual third party that can be trusted for correctness and availability. The blockchain are immutable and hence any operation that alters the data in any block violates the blockchain consensus[4] rule and are rejected by the blockchain network.

In this paper, we propose a digital voting system that is completely built over blockchain protocol. Our voting system does not depend on a centralized trusted party for ballot tallying or result publishing. It is also a platform-independent system and provides comprehensive security assurances with the help of various cryptographic techniques. In a nutshell, this system is going to overcome many of the issues faced by current voting system.

The rest of the paper is organised as follows:
Section II consists of literature review, Section III describes Statement of problem and proposed solution, Section IV is Results and Discussion, Section V gives idea about future scope of this system, Section VI enlists all the references.

II. Literature Review

A. Voting systems in public bulletin[10] :

In the following, we outline the key cryptographic techniques used in public bulletin based voting systems

Homomorphic Encryption: Homomorphism feature enables one to work on cipher-texts without unscrambling them. For a casting a ballot framework, this property enables the scrambled polls to be tallied by any outsider without releasing any data in the vote.

Mix-net: Mix-net was proposed in 1981 by Chaum. The fundamental thought of mix-net is to play out a re-encryption over an arrangement of ciphertexts and shuffle the request of those ciphertexts. Mix node just knows the node that it quickly got the message from and the prompt goal to send the shuffled messages to.

Zero-knowledge proof: Zero-knowledge evidence is frequently utilized in a casting a ballot framework to let the prover to demonstrate that the announcement is surely what it asserted without uncovering any extra knowledge of the announcement itself. In a voting system, the voter should convince the authority that his ballot is valid by proving that the ballot includes only one legitimate candidate without revealing the candidate information.

B. Voting Systems in Blockchain:

The blockchain-based voting systems can be discussed under three broad categories as follows.

Voting systems using cryptocurrency: In [5], authors propose a voting system based on Bitcoin. In their voting system, the ballot does not need to be encrypted/decrypted as they employ the protocol for lottery. Random numbers are used to hide the ballot that are distributed with the help of zero-knowledge proof. Making deposit before voting may keep the voters to comply with the voting protocol while the malicious voters can still forfeit the voting by refusing to vote.

Voting systems using smart contract: In [6], it is an open voting network which is a decentralised and self-tallying Internet voting protocol with maximum voter privacy using Blockchain. They have incorporated smart contract as a public bulletin to achieve self-tallying. But this voting system can only work with two candidates voting (yes/no voting) and the limitation of voters makes it impractical for a real large scale voting system.

Voting systems using blockchain as a ballot box: Here the blockchain structure is actually acting as a ballot box for the system. the votes are stored inside the blocks with the help of encryption techniques. This provides more robust and flexible system.

III. Statement of Problem and Proposed Solution

Problem Statement: To develop a digital decentralized fault tolerant voting application based on Blockchain architecture and consensus protocol which will be more secure, robust and more stable than the current operating voting system.

Proposed System Architecture:

A. Parts of the System:

- **Peer:** It is the machine located at a polling booth which will act as a node connected to distributed the peer to peer network of our voting system.
- **Voter:** It is a actual citizen of India who is going to cast the vote into the system from any one of the peer node of the distributed network.
- **Authentication Server:** This is the server which will authenticate the validity of voter. In proposed system, the authentication server is UIDAI Aadhaar Server of Government of India.
- **Admin:** These are the officials of Election Commission of India, having rights of ballot ballot tallying and result publishing.
- **Blockchain System:** This is the system on which actual voting takes place. Every peer in the network holds the copy of blockchain locally. The tasks of proof of work, consensus and byzantine fault tolerance is carried out by the blockchain system.

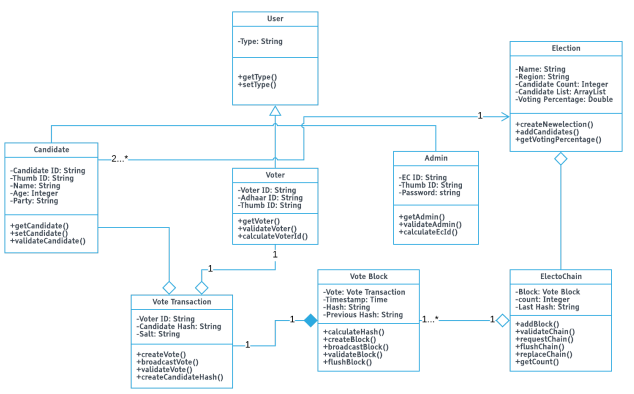

Fig 3.1: Class Diagram of the proposed system

B. System Architecture:

- **Voter Authentication:** Voter authentication is carried out using Aadhaar API services. The system uses two different Aadhaar API for authentication. Voter has to provide his thumb impression which is validated by Aadhaar Thumb Impression, API followed by this, voter has to provide his/her Aadhaar ID which is again validated by Aadhaar eKYC API. Here all the details of voter like age, constituency, etc are also verified by Aadhaar API and then the system creates a unique, secure voter ID by using SHA-256[7] hashing algorithm.

- **Vote Casting and Transaction Broadcasting:** After the successful authentication voter has to cast his/her vote for any one of the candidate from the candidate list. When voter selects any one of the candidate, a vote-transaction is created at that peer where vote has been casted. This vote-transaction consists of following fields-

    1) *Transaction ID:* A 256-bit unique ID to identify the transaction.
    2) *Voter ID:* A 256-bit secure hash ID of the voter created at the time of authentication.
    3) *Salt:* A random string of characters generated dynamically which is appended on candidate ID.
    4) *Candidate Hash*: Every candidate is assigned with a unique candidate ID by Election Commission of India. A salt generated is appended on this candidate ID and this complete string (ID + salt) is hashed using SHA-256 algorithm. This unique hash acts as candidate Hash.

This generated vote-transaction is broadcasted to every peer in the distributed peer to peer network. At the time of broadcasting, the vote-transaction is encrypted using SSL[8] encryption technique. The transaction is decrypted when received by each peer.

- **Proof of Work and Block Broadcasting:** When the broadcasted transaction is received by each peer, they start the proof of work process. This process is basically to create a block for the vote-transaction. Every block contains only one vote-transaction. Here every peer will try to solve a cryptographic mathematical puzzle having some difficulty count. The peer who solves this puzzle first creates a hash ID for the block. This block consists of following fields:
    1) *TimeStamp:* It is an instance of time at which the block is created. It is mentioned in milliseconds.
    2) *Transaction:* This is the vote-transaction created earlier.
    3) *Hash ID:* This is a secure 256-bit hash ID created by a proof of work having difficulty count 4 and a random nonce string.
    4) *Previous Hash ID:* This is the 256-bit hash ID of the previous block in the blockchain.

After the block is created, the peer who has created the block again broadcasts it to every other peer in the distributed P2P network. After receiving this broadcasted block every peer stops their proof of work process and start the next process of block validation.

- **Block Validation and Consensus Protocol:** After receiving the block, every peer starts the process of Block validation. Here starting from the new block to the genesis block (first block of blockchain) every peer checks whether previous hash ID of each block matches with hash ID of previous block in their local copy of blockchain. If all the matches are true then the block validation process has succeeded for that peer and then that peer broadcasts a success packet into the P2P network. If any of the match gets false, then the block validation fails at that peer and then that peer broadcasts a failure packet into the P2P network. Now by applying the consensus protocol, if the majority (more than 50%) of the peers is having success then only the new block will get added to the blockchain copy at each peer. If majority peers is having failure in block validation, then the new block is discarded and flushed out of the system. Hence, every time rather than trusting only one server, here we are taking the majority by considering each peer thereby increasing the reliability upto great extent Also the voter is given a chance to cast his vote again. If the block is successfully added to each blockchain copy at each peer, then the confirmation is sent to the voter.

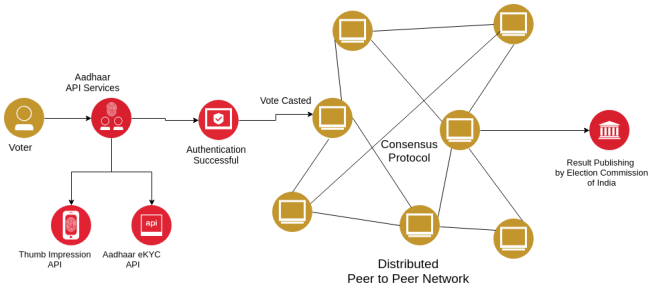

Fig 3.2: Architecture Control Flow of proposed System

- **Byzantine Fault Tolerance[9]:** Everytime after the consensus protocol, the process of byzantine fault tolerance is carried out. Here, if the blockchain copy at any peer is corrupted or disturbed or tampered, then that peer requests the copy of blockchain from its nearest peer holding valid blockchain copy, and replaces its own blockchain copy by the new one.
  Consider the scenario, if 80% of the total peers is having success in block validation process, then according to consensus, the block will be added to every blockchain copy. But the remaining 20% peers having the failure at block validation, will get to know that there is some fault in there blockchain copy. So they will go for the byzantine fault tolerance and they will replace their faulty blockchain copy with the valid one. Hence, after every vote casted, the system again comes to the valid, stable and consistent state.

- **Vote Calculation and Result Publishing:** This phase of system is only reserved for the officials of Election Commission of India. Here the officials are authenticated by using their thumb impression and election-commission ID. At the time of vote

calculations, The transactions in every block is traced out. The salt from each transaction is appended to the each candidate ID and hash is generated by SHA-256. If the newly created hash and the candidate Hash from the transaction matches then the vote is casted for that candidate for which we got the successful hash-match and we increment its vote-count by 1. In this way result is calculated and can be published in a very short time. Also as every peer holds the copy of same blockchain, hence the result can be calculated at any peer. Also the total number of blocks gives the total number of votes casted, hence the percentage of voting can be easily recorded.

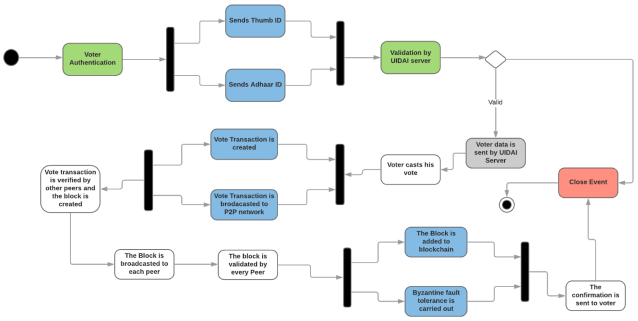

Fig 3.3: Operational Flow of Proposed system

IV. Results and Discussion

The system proposed in this paper eradicates most of the issues faced by the current voting system in India. The issues overcame by the proposed systems are as follows:

1) Privacy of the voter is completely maintained as the voter's details are never kept at any peer they are directly verified at UIDAI Aadhaar API servers. 2) As every piece of data is stored using hashes and salts generated by SHA-256, the anonymity of the voter and also the data security is maintained at its peak. 3) Authentication is carried out by thumb impression and Aadhaar ID with secure UIDAI servers hence, fake voting and double voting is avoided. 4) The time and expenses required to carry out election is remarkably lesser than current voting system. 5)The confirmation of the successful vote cast is provided to the voter. 6) The results can be published in a very short time.

Hence, the proposed system is better than the current voting system in each and every aspect. The reliability, data security and the transparency achieved is remarkably high.

V. Future Scope

In this paper, this system is not offering "vote from home" facility, as there are major issue with this thing. There may be the situation of threatening or manipulation by some political parties which may seriously harm to the integrity of election procedure. Hence, to avoid this issue and to provide facility of "vote from home", an emotion detection module can be incorporated by using Image processing and machine learning algorithms. By this, the facial emotions of the voter at the time of voting can be detected with the help of WebCam and if the emotion statistics are bending more towards fear, anger or anxiety, etc. then the voter's session will be blocked or halted for some time and he will be asked to vote after some time.

One more improvisation can be done in terms of data storage. Some cloud computing services like AWS, Microsoft Azure, etc. or the services like InterPlanetary File System can be used to store the data which is currently stored locally at each peer. This will also help in increasing the scalability of application upto greater extent.

VI. References

[1] Blockchain Wikipedia https://en.wikipedia.org/wiki/Blockchain

[2] Satoshi Nakamoto Wikipedia https://en.wikipedia.org/wiki/Satoshi_Nakamoto

[3] Peer to Peer Network https://www.computerworld.com/article/2588287/networking/networking-peer-to-peer-network.html

[4] Consensus Protocol https://blockgeeks.com/guides/blockchain-consensus/

[5] Zhao, Z., Chan, T.H.H.: How to vote privately using bitcoin. In: International Conference on Information and Communications Security. pp. 82–96. Springer (2015)

[6] McCorry, P., Shahandashti, S.F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy. IACR Cryptology ePrint Archive 2017, 110 (2017)

[7] SHA-256 : https://www.mycryptopedia.com/sha-256-related-bitcoin/

[8] SSL Encryption: http://info.ssl.com/article.aspx?id=10241

[9] Byzantine Fault Tolerance https://medium.com/loom-network/understanding-blockchain-fundamentals-part-1-byzantine-fault-tolerance-245f46fe8419

[10] Platform-independent Secure Blockchain-Based Voting System, Monash University Australia CSIRO,Australia,The Hong Kong Polytechnic University https://eprint.iacr.org/2018/657.pdf

[11] followmyvote https://followmyvote.com/

[12] Blockchain Voting System, Sagar Shah, Qaish Kanchanwala, Huaiqian Mi, from Northeastern University, https://www.economist.com/sites/default/files/northeastern.pdf